4. 16. To construct an Optimal Binary Search Tree (OBST) for the given set of keys and their frequencies, and to display the cost table, root table, and minimum OBST cost.

**AIM**

To implement the Optimal Binary Search Tree algorithm for the given keys and frequencies, construct the OBST, and display the cost and root matrices.

**ALGORITHM**

1.Start

2.Input number of keys n, the keys, and their frequencies.

3.Initialize:

* cost[i][i] = freq[i] (cost of a single key).
* root[i][i] = i (root of a single key is itself).
* w[i][j] = sum(freq[i..j]).

4.For subsequence lengths 2…n:

* For each interval (i,j), compute the total weight w[i][j].
* For each possible root r in [i..j]:
* cost[i][j] = min( cost[i][r-1] + cost[r+1][j] + w[i][j] )
* Update root accordingly.

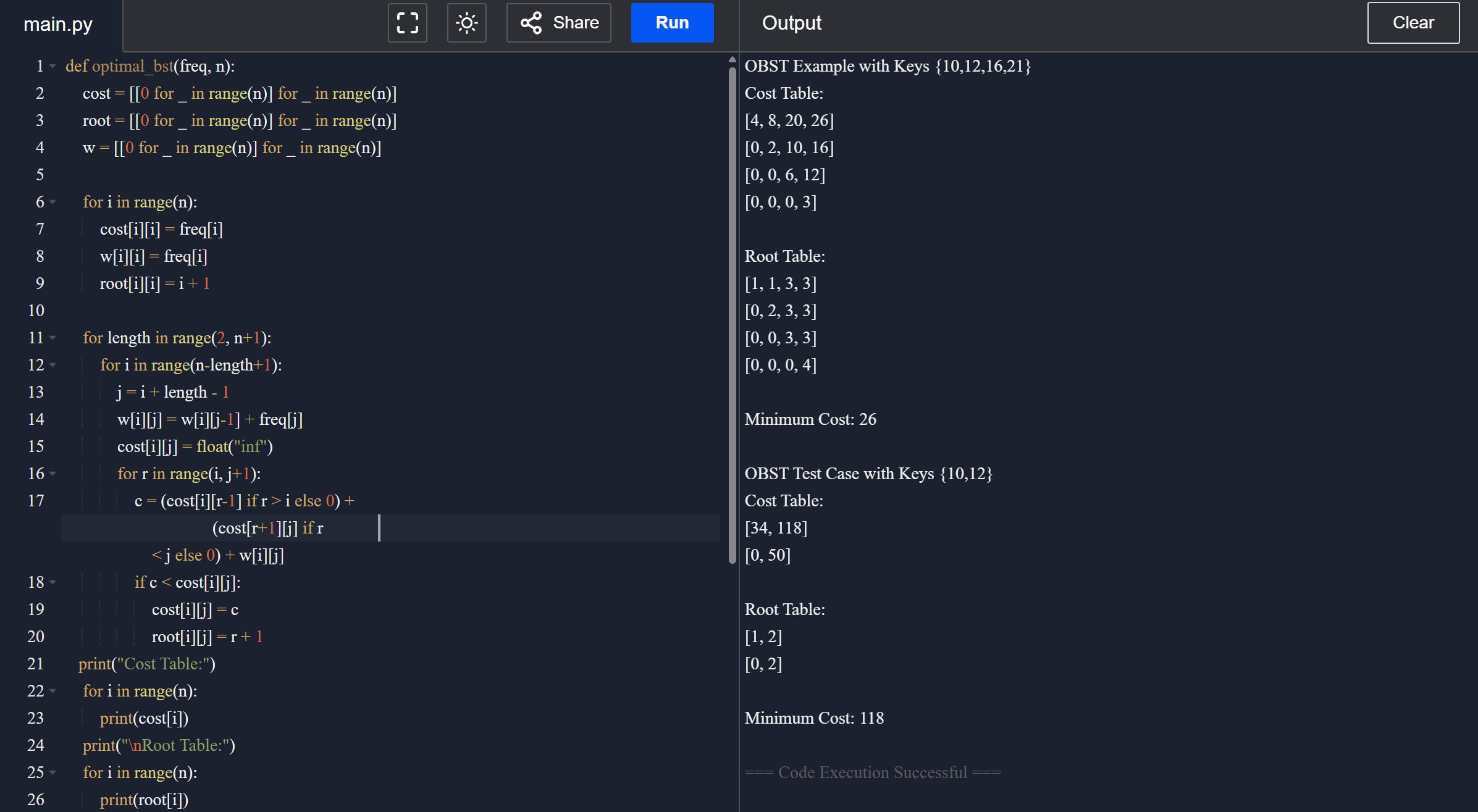
5.The minimum OBST cost = cost[0][n-1].

6. Display the cost and root tables.

7.End

.

**PROGRAM**



Input:

N = 4

Keys = {A, B, C, D}

Frequencies = {0.1, 0.2, 0.4, 0.3}

**RESULT:**

* For **Keys {10,12,16,21}**, the OBST cost = **26**.
* For **Keys {10,12}**, the OBST cost = **118**.  
  The program successfully constructs the OBST and displays the cost and root tables.

**PERFORMANCE ANALYSIS:**

* **Time Complexity: O(n³)**
* **Space Complexity: O(n²)**